Asleep Alpacas: Justin Zou (Piggy), Wen Hao Dong (Jal Hordan), Yoonah Chang (Yelena), Liesel Wong (King Hagrid)

SoftDev

P00 -- Move Slowly and Fix Things/Creating a website/

2021-10-27

time spent:

Scenario One: Your team has been contracted to create a collaborative storytelling game/website.

- Users will have to register to use the site.
- Logged-in users can either start a new story or add to an existing story.
- When adding to a story,
  - Users are shown only the latest update to the story, not the whole thing.
  - A user can then add some amount of text to the story.
- Once a user has added to a story, they cannot add to it again.
- When creating a new story,
  - Users get to start with any amount of text and give the story a title.
  - Logged in users will be able to read any story they have contributed to on their homepage (the landing page after login).

# **Database Organization:**

## stories.

id (INTEGER)	author_id (INTEGER)	title (TEXT)	full_story (TEXT)	last_update (TEXT)
--------------	------------------------	--------------	----------------------	-----------------------

### users:

id (INTEGER)	username (TEXT)	password (TEXT)
--------------	-----------------	-----------------

#### contributions:

user_id (INTEGER)	story_id (INTEGER)
-------------------	--------------------

## **Components and Purpose:**

- Templates
  - home.html: Displays a list of the logged in user's stories they contributed to
  - login.html: Contains a form with username and password input
  - register.html: Contains a form with username and password input
  - new.html: Contains a form with the story title and starting body
  - discover.html: Displays a list of stories the user hasn't contributed to
  - view.html: Display a full story and its title

- edit.html: Displays the latest update and has a form containing a textarea tag to continue the story
- Flask App
  - "/": Renders home.html
  - "/logout": Pops user id session if exists and redirects to "/login"
  - "/login": Handles logging in
  - "/register": Handles creating new users
  - "/create": Add new stories
  - "/discover": Lists stories the user hasn't contributed to
  - "/story/<story\_id>": Shows the entire story or allows a user to contribute to the story depending on whether or not they've already contributed to the story
- SQLite database
  - Relational database containing tables of stories, users, and contributions (a user and which story they've contributed to)jsdf

# **Template Representation:**

Home Page:

Story New Discover Login
Stories story! by author!
+ e × t
Story 2 by author
r

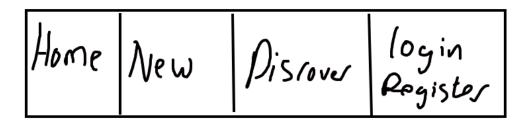
Lagin	Dogo:
1.02111	Page:

	Login
Username:_	
Password:	

Register

Register Page:	New Story:	
Register	Create Title:	
Username:Password:	Text:	
Discover:	View:	
Discover  Story 1 by [Author] [Latest Update Text]  Story 2 by [Author] [Latest Update Text]  Story 3 by [Author] [Latest Update Text]	Create  Title: by:  Story:	
	Edit Page:  Edit  Title: by: Last Update: Text:	

Linkage: Each of the tabs seen on the homepage will go to their respective page. I.e. New goes to create a story page, discovery goes to discovering page etc. Clicking on a story with the view button from the discovery page will lead to a view page and clicking on the edit page will go to that edit page as well. The User would be able to go back with a navbar like this:



The login and register would change to logout if they are logged in already. The Register, Login, Edit and Create pages will all have a submit button and be submitted through the enter key.

## **Division of Tasks:**

Templates (Front End): Yoonah

SQLite database: Justin

Flask App and Python: Liesel, Wen Hao

#### **Components Map:** HOME.html logout Displays contributions table/ user stories if logged in REGISTER.html NEW.html LOGIN.html DISCOVER.html Goes to edit.html which Contains flask form with Contains flask form with Display stories table Contains a new flask form new username and username and password with title and text of a story password Sqlite checks User clicks on password story Flask access input through Flask access input through a function a function Sqlite inserts input into Sqlite inserts input into stories table users table Sqlite inputs the story\_id and user\_id into the contributions table view.html Edit.html Goes to view.html which Flask access input through access stories table and a function displays title and full story Sqlite inserts input into last\_update in stories table Sqlite inputs the story\_id and user\_id into the contributions table

The solo Logout Component is to signify that you can logout from any of the pages seen.

# Site Map:

